A Server Allocation and Placement Algorithm for Content Distribution

Craig Cameron, Steven H. Low, and David X. Wei
Depts. of Computer Science and Electrical Engineering
Caltech, Pasadena, USA
{cwcam,slow,weixl}@caltech.edu

Abstract — **It is well known that optimal server placement is NP-hard. We present an approximate model for the case when both clients and servers are dense, and propose a simple server allocation and placement algorithm based on high-rate vector quantization theory. The key idea is to regard the location of a request as a random variable with probability density that is proportional to the demand at that location, and the problem of server placement as source coding, i.e., to optimally map a source value (request location) to a codeword (server location) to minimize distortion (network cost). This view has led to a joint server allocation and placement algorithm that has a time-complexity that is linear in the number of clients. Simulations are presented to illustrate its performance.**

I. Introduction

A content distribution network reduces propagation delay, relieves server load, balances network traffic, improves service reliability, and disperses flash crowds. Content from a provider is distributed to multiple servers in the network, and a client request is served by a 'nearest' server. Here, proximity may refer to geographical distance, hop count, network congestion, server load or a combination. A central design issue is how to allocate and place servers in the network.

Server placement is known as the K-median problem in graph theory: given a graph with N nodes, each node i with a request rate $r(i)$, pick $K(< N)$ nodes as servers and assign each node to one of these servers so that the total weighted distance between all nodes i and their servers, weighted by $r(i)$, is minimized. This problem is shown in [3, 5] to be *NP*-hard for general graphs.

In this paper, we take a completely different approach, focusing on the case where both client and server densities are high. In this regime, server placement can be regarded as a high-rate vector quantization problem. *The key idea is to regard the location of a request as a random variable with a probability density that is proportional to the demand at that location, and the problem of server placement as source coding, i.e., to optimally map a source value (request location) to a codeword (server location) to minimize distortion (network cost).* This view has led to a simple *joint* server allocation and placement algorithm with time complexity linear in NM where N is the number of clients (e.g., client side proxies) and M is the number of content providers; in particular, it is linear in N. Preliminary simulation results suggest that it has a good performance-complexity tradeoff.

[1]This work was supported by Caltech Lee Center for Advanced Networking and Yuen Research Fund.

II. High-density model

We start with the case of a single 'website', and extend it to the case of multiple 'websites'. A 'website' in our model may represent information produced by a subset of sensor nodes, a content provider, an entire website, a collection of files or applications, or a single file or application. A 'node' may represent another sensor node, an end user of the website, a client-side proxy that serves a family of end users in the same local area network or same organization. By placing a server 'at a node', we mean placing a server 'near' the end user or client-side proxy represented by the node, e.g., on the same subnet.

A Single website

Let *every* point $z = (z_1, z_2) \in \Re^2$ be a node. The request rate of node z is $r(z)$ requests per minute. Interpret the normalized request rate

$$f(z) \quad = \quad \rho^{-1} r(z) \quad \text{where} \quad \rho := \int r(z)\, dz$$

as the *spatial density* of requests. The goal is to place K servers at locations $s = (s_1, \ldots, s_K)$ so as to minimize total network cost of serving the requests, defined as follows.

Let $d(z, s_k)$ be the 'distance' of serving a request from node z by the kth server located at node s_k. Given server locations s, the distance measure $d(z, s_k)$ partitions V into what are called Voronoi cells $V_k \subseteq V$ defined by:

$$V_k \quad = \quad \{z \mid d(z, s_k) \le d(z, s_l), \ \ \forall l\}$$

Hence members of Voronoi cell V_k are *nearest neighbors* of server at s_k. The cost of serving a Voronoi cell is $\int_{V_k} r(z) d(z, s_k) dz = \rho \int_{V_k} f(z) d(z, s_k) dz$. When there are K servers at positions s, the network cost is defined as

$$c(K, s) \quad := \quad \rho \sum_{k=1}^{K} \int_{V_k(s)} f(z) d(z, s_k)\, dz$$

Our goal is to choose server locations s so as to minimize $c(K, s)$. Since this is NP-hard, we seek simple algorithms with good performance.

The idea is to regard the location Z of a request as a random variable with probability density f, and the problem of server placement as source coding, i.e., to optimally map a source value (request location Z) to a codeword (server location s) to minimize distortion (network cost). Our main assumption, which is valid when both client and server densities (large K) are very high, is that $V_k(s)$ is very small, and that $f(z)$ is smooth so that $f(z) \simeq f(s_k)$ over $V_k(s)$.

We choose the following distance measure:

$$d(z, s_k) \quad := \quad \alpha(s_k)||z - s_k|| + \beta(s_k) \tag{1}$$

with the following interpretation. The first term models the delay between client location z and server location s_k and the second term the server load. The implicit assumption is that the delay is proportional to geographical distance $||z - s_k||$ and that the proportionality constant $\alpha(s_k)$ captures queueing delay due to congestion (e.g., total delay is 2 times propagation delay). The parameter $\alpha(s)$ is assumed uniform over the small region $V_k(s)$ but can vary across Voronoi cells. In a wireless setting, the geographical distance $||z_s - k||$ is also a measure of required transmit power, either via multi-hop relay or single-hop broadcast. The second term assumes that the server load depends only on location s_k, e.g., $\beta(s_k)$ can be inversely proportional to the server capacity at s_k, or proportional to the common request density in region $V_k(s)$ which decreases with the total number K of servers, e.g., $\beta(s_k) \propto f(s_k)/\sqrt{K}$.

We specify server location in this continuum model by *server density* $\lambda(z)$, with the interpretation that the *fraction* of servers in an infinitesimally small area dz around z is $\lambda(z)dz$. Hence the *number* of servers in any region A is $K \cdot \int_A \lambda(z)dz$. Note that $\int \lambda(z)dz = 1$ so λ can also be regarded as the probability density of server location. Our goal is to determine the optimal server density $\lambda^*(z)$ that minimizes the (approximate) network cost $C(K, s)$.

With this formulation, we can show that the 'optimal' server density $\lambda^*(z)$, within the high-density model, is

$$\lambda^*(z) \quad = \quad \left(\frac{\hat{f}(z)}{||\hat{f}||_{2/3}} \right)^{2/3} \tag{2}$$

and it incurs an 'optimal' cost, with K servers, of

$$c^*(K) \quad = \quad \frac{2\rho}{3\sqrt{\pi}} \cdot \frac{||\hat{f}||_{2/3}}{\sqrt{K}} \; + \; \rho \cdot E\beta(Z) \tag{3}$$

where the expectation $E(\cdot)$ is taken with respect to distribution f. Here, $\hat{f}(z) := \alpha(z)f(z)$ and $||\hat{f}||_p$ is the L_p norm, $0 < p \le \infty$, defined as $||\hat{f}||_p = \left(\int \hat{f}(z)^p \, dz\right)^{1/p}$.

We interpret these preliminary results.

Remarks:

1. Expression (2) says that the optimal server density $\lambda^*(z)$ is proportional to the 2/3-root of the weighted spatial density $\alpha(z)f(z)$ of requests. This highlights the importance of spatial density of requests in server (or file) placement and agrees with intuition: more servers should be placed where request concentrates. The 2/3-root comes from the choice of Euclidean distance and Hölder's inequality (it will be square root if $d(z, s_k)$ is *squared* Euclidean distance).

2. Expression (3) says that the cost under $\lambda^*(z)$ is proportional to the total volume ρ of requests, and decreases with the number K of servers as $1/\sqrt{K}$. It also increases with the L_p norm of the weighted spatial density $\alpha(z)f(z)$ and the expected server load $E\beta(Z)$.

3. We emphasize that the original problem is NP-hard, and these results are for an approximate model for the case where both client and server densities are high. Unlike the previous approaches that produce only numerical algorithms that provide no insight on the role of various parameters, the high-density approximation leads to a clear and intuitive role in server placement for these parameters.

4. Expression (2) for $\lambda^*(z)$ suggests a server placement strategy where server density is proportional to the 2/3-power of the request density, $f(z)^{2/3}$, or equivalently, of the request rate, $r(z)^{2/3}$.

B Multiple websites

Consider J websites indexed by $j = 1, 2, \ldots, J$. Suppose requests to website j has a total volume of ρ_j and a spatial density $f_j(z)$ (or equivalently, a request rate $r_j(z) = \rho_j f_j(z)$). Out of a total of K servers, k_j servers are allocated to serve website j such that $\sum_{j=1}^J k_j = K$. We assume that $\beta_j(s_k)$ in the definition (1) of distance is $\beta_j(s_k) = \beta_j/\sqrt{k_j}, \forall k$. The k_j servers are placed according to the optimal server density λ_j^* so that the cost associated with website j is $c_j(k_j) = \gamma_j\rho_j/\sqrt{k_j}$ where

$$\gamma_j \quad := \quad \frac{2}{3\sqrt{\pi}} ||\alpha_j f_j||_{2/3} \; + \; \beta_j \tag{4}$$

and

$$||\alpha_j f_j||_{2/3} \quad := \quad \left(\int (\alpha_j(z) f_j(z))^{2/3} dz \right)^{3/2}$$

as explained in the last subsection. Note that servers for different websites can be co-located at the same node. We will choose server allocation k_j to minimize the network cost:

$$\begin{aligned} c^*(K) \quad &= \quad \min_{k_j} \sum_j c_j(k_j) = \sum_j \frac{\gamma_j \rho_j}{\sqrt{k_j}} \\ \text{s. t.} \qquad & \sum_j k_j = K, \quad k_j \in \{0, 1, \ldots, K\} \end{aligned}$$

For large K, relax the constraint that k_j be integers. We hence solve the following simple convex program:

$$\begin{aligned} c^*(K) &= \min_{k_j} \sum_j \frac{\gamma_j \rho_j}{\sqrt{k_j}} \\ \text{s. t.} \quad & \sum_j k_j = K, \quad k_j \geq 0 \end{aligned}$$

The optimal allocation and cost are:

$$k_j^* = \frac{(\gamma_j \rho_j)^{2/3}}{\sum_l (\gamma_l \rho_l)^{2/3}} \cdot K \tag{5}$$

$$c^*(K) = \left(\sum_j (\gamma_j \rho_j)^{2/3} \right)^{3/2} \cdot \frac{1}{\sqrt{K}}$$

where γ_j are given by (4).

Remarks:

1. Recall that ρ_j represents the popularity of website j, and f_j represents the spatial density of requests for website j. They are related through the request rate $r_j(z) = \rho_j f_j(z)$. Hence, optimal allocation depends critically on website popularities as well as spatial densities of requests. Specifically, the fraction of servers allocated to website j should be proportional to $(\gamma_j \rho_j)^{2/3}$. The optimal cost is proportional to $\left(\sum_j (\gamma_j \rho_j)^{2/3}\right)^{3/2}$ and inversely proportional to $\sqrt{K}$.

2. We can combine equations (2) for optimal placement and (5) for optimal allocation to express the optimal number of servers for each website j in terms of the total number K of servers, as $\lambda_j^*(z) k_j^*$. For instance, for $\alpha(z) = 1$ and $\beta(s_k) = 0$,

$$\lambda_j^*(z) k_j^* = \frac{r_j(z)^{2/3}}{\sum_l \int r_l^{2/3}} \cdot K$$

Hence, the optimal density is proportional to $r_j(z)^{2/3}$, as a fraction of total request rate for *all* websites.

III Algorithm and performance

Recall that ρ_j represents the popularity of website j, and f_j represents the spatial density of requests for website j. They are related through the request rate $r_j(z) = \rho_j f_j(z)$. The preliminary results discussed in the last subsections highlight the importance of spatial distribution of requests in server placement and website popularity in server allocation that agrees with intuition: more servers should be allocated to more popular websites (with larger ρ_j), and these servers should be placed where requests concentrate. Moreover, they suggest that website j should be allocated servers proportional to $(\gamma_j \rho_j)^{2/3}$ (equation (5)), where ρ_j represents website popularity and γ_j captures the spatial density and cost metrics. These servers should be placed with a server density proportional to $(\alpha(z) f_j(z))^{2/3}$ (equation (2)).

As mentioned above, the spatial distribution of requests is not well exploited in current systems, both because of the difficulty in measuring it empirically [4, 2, 6], and because of the lack of a theoretical understanding of its role. Our model can help focus future effort to address critical problems.

Based on these insights, we have derived a (discrete) graph algorithm that jointly allocate and place servers [1]. It has a time complexity that is linear in the number N of servers. In preliminary simulations, with the number of nodes N ranging from 100 to 20,000, suggest that it consistently achieves a cost that is about 1.5 times the cost of best approximation (K-median) algorithm. The K-median algorithm solves instances up to $N = 1,000$ (with running time of 294 sec on $N = 1,000$ on 1.5GHz Pentium 4 processor with 256Mb RAM) whereas our algorithm can solve instances larger than $N = 20,000$ (with running time of 0.69 sec on $N = 20,000$ on the same machine). This may be an appropriate tradeoff for large-scale self-organizing networks we envision.

References

[1] Craig W. Cameron, Steven H. Low, and David X. Wei, "High-density model for server allocation and placement," In *Proceedings of ACM Sigmetrics*, Marina del Rey, CA, June 2002.

[2] J. Kangasharju, K. W. Ross, and J.W. Roberts, "Locating copies of objects using the domain name system," In *Proceedings of the 4th International Caching Workshop*, March 1999.

[3] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. II: the p-medians," *SIAM J. Appl. Math.*, 37(3):539–560, December 1979.

[4] Balachander Krishnamurthy and Jia Wang, "On network-aware clustering of web clients," In *Proceedings of ACM SIGCOMM 2000*, August 2000.

[5] C. Papadimitriou, "Worst-case and probabilistic analysis of a geometric location problem," *SIAM J. Comput.*, 10:542–557, 1981.

[6] A. Shaikh, R. Tewari, and M. Agrawal, "On the effectiveness of DNS-based server selection," In *Proceedings of IEEE Infocom*, April 2001.